Rev. 1.1, 6/2003

Building an NFS DHCP/BOOTP Server for Use With Sandpoint and MVP Linux

Maurie Ommerman
CPD Applications
risc10@email.mot.com

This document describes the steps necessary to configure a Linux DHCP/BOOTP server for use with the MontaVista Hard Hat Linux™ CDK, V2.0, running on either a Sandpoint or MVP evaluation board client. PowerPC™ microprocessors are supported on the Sandpoint and MVP evaluation boards.

This document covers the following topics:

CAUTION

Do not copy IP addresses verbatim from this document; use a correct IP address for your network.

Do not have more than one DHCP server active per network.

Refer to Part VI, “Dynamic Host Configuration Protocol (DHCP).”

NOTE

The MontaVista Hard Hat Journeyman Edition does not fully support Yellow Dog Linux™ as a host. Although it can be set up as a DHCP/BOOTP server, and perform an NFS boot, the resulting root file system will be read-only. Because you can not change the files, useful work cannot be accomplished on this platform and host combination. However, the full MontaVista release does support Yellow Dog Linux as a host. I have not tested this full version.

On the other hand, Red Hat™ V7.0, which is supported, and Red Hat™ V7.2, which is not supported by MontaVista, work correctly. The NFS boot mounts the root file system read/write, permitting the performance of useful work on both of these platform and host combinations.

Part I Introduction

This document is organized by parts that explain the steps required [summarized in Section 2.2, "Required Server Steps"] to configure a Linux system for serving NFS remote boot from a DHCP/BOOTP server.

- Part I, "Introduction," gives an overview of this document and defines terminology used.
- Part II, "General Methodology," provides an overview of client and server interaction and summarizes the steps in the configuration process.
- Part III, "Server Machine," discusses the server development environment and the required Red Hat package manager (RPM) compilers and utilities.
- Part IV, "Installing the MontaVista CDK V2.0," provides general information about installing the CDK V2.0. and indicates where instructions can be found to produce the necessary kernel for the PowerPC architecture. This section also indicates directory locations.
- Part V, "Linux Boot Methods," outlines common methods of booting Linux, depending on the hardware used.
- Part VI, "Dynamic Host Configuration Protocol (DHCP)," repeats a critical notice regarding IP addresses and servers. It also outlines how to set up and configure DHCP, including:
  - The necessary code for the DHCP daemon configuration file
  - A line-by-line explanation of that code
  - Instructions for restarting the DHCP daemon
- Part VII, "Trivial File Transfer Protocol (TFTP)," addresses the trivial file transfer protocol (TFTP) requirements based on the Linux software release used. Also included is the code for the recommended TFTP file, with a line-by-line explanation of the code, instructions for creating a symbolic link, and restarting the TFTP.
- Part VIII, "Set up Network File System," discusses how to set up the network file system (NFS) export files, provides recommendations on code to be added to the export files, and outlines procedures for controlling the NFS daemon.
- Part IX, "Set Up resolv.conf File," provides directions to build the resolv.conf file, which performs domain name resolution. The recommended code is included with a line-by-line explanation.
- Part X, "Other Problems," addresses how to resolve a common problem, shell start failure. A link is provided for the /opt/hardhat/devkit/ppc/82xx/target/bin directory.
- Part XI, "Host Ethernet Diagnostic Commands," gives instructions for:
  - Checking the status of gateway and network setup

- — Viewing network traffic
- — Using the message log

- Part XII, “Starting Linux with the Executable,” addresses downloading the executable. It also provides examples of both the MVP and Sandpoint splash screens, as well as a sample Sandpoint session.
- Part XIII, “Revision History,” provides the document history.

1.1 Terminology

The following terms are used in this document.

- Bash shell—Extension to the Bourne shell, which is popular on Linux systems and is sometimes called GNU Born Again Bourne Shell. This is the default shell for most Linux systems because Linux uses GNU tools exclusively.
- Boot—Program that begins at hardware reset, which prepares the hardware for loading an OS.
- BOOTP—Short for bootstrap protocol, this is a broadcast request from a client to a BOOTP server, which is usually a DHCP server. If the server determines that the client is valid, an IP address is returned to the client, permitting the boot process to continue.
- Broadcast—Sends a message to all listeners on an Ethernet or subnet.
- Client—A program or machine that makes requests to another machine called a server.
- CDK—Cross Development Kit is a set of development tools for Intel- or Mac-based Linux machines to build PPC Linux objects.
- Daemon—A Linux and Unix term that refers to a small program that runs in the background and is owned by root. A daemon listens for activity, which it can respond to and then performs services for that activity. Daemons pertinent to this document include inetd, xinetd, and dhcpd.
- DINK32—Small OS debugger for the Sandpoint evaluation board.
- DHCP—Dynamic host configuration protocol used by a server to respond to client requests for automatic assignment of IP addresses.
- dhcpd—This daemon performs the DHCP service, monitoring ethernet traffic to identify incoming requests for IP address assignment and responding to these requests if they are validated.
- FTP—File Transfer Protocol, used for tranferring files between machines on a network.
- Hard Hat Linux—Also called hhl, a product of MontaVista Software.
- Host—A machine that can be used to build kernels. The host may or may not be the same architecture. For example, Freescale builds kernels on both a G4 machine and a Intel machine.
- IDE—Integrated Device Electronics is the standard interface for many devices including hard drives, CD-ROM drives, and others.
- inet, inetd—inet is the Internet service performed by inetd, a daemon that monitors ethernet traffic to identify incoming traffic destined for the IP address on which the daemon resides. This daemon is supported by Mandrake Linux V7.0.
- IP Address—Internet Protocol address, a quartet of digits that represent the addressing scheme and identify the packet format. The IP address is usually represented by decimal numbers that specifically identify a node in a network. Each entry of the quartet has a value from 0 to 256. The representation is xxx.yyy.zzz.aaa.
- Kernel—The central module of OS that interfaces the hardware to the software.

- Mandrake Linux—Product of MandrakeSoft, Inc. (designer and distributor of desktop Linux sources and solutions).
- MAC—Media Access Control is the physical address of an ethernet card. Represented by a sextet of hex values, each member has a value from 00–ff. The first three members are assigned to a manufacturer, the last three members are uniquely assigned by the manufacturer. The representation is xx:yy:zz:aa:bb:cc.
- MontaVista Software—One of several companies that modifies and distributes Linux sources for embedded applications.
- MVP —An evaluation board, with a Galileo network interface, that uses two MPC7450 processors and is capable of running an SMP Linux.
- NFS—Network File System, a protocol used to share and access files over a network regardless of machine, operating system, or architecture.
- OS—Operating system.
- RAM disk—Random Access Memory that has been configured to simulate a disk drive.
- Red Hat—Red Hat, Inc. is one of several companies that uses the freely distributed desktop Linux sources and packages them for distribution. They developed the RPM which has become a standard for Linux.
- Red Hat package manager (RPM)—Program to install and to maintain groups of Linux binary, libraries, documentation, and other Linux-style objects.
- Required files—DHCP/BOOTP servers rely on certain files that define the actions of these servers.
- Server—A machine that services requests from clients.
- SMP—Symmetric multiprocessing program, a computer architecture that features a single operating system that uses two or more processors.
- Srecord or srec—A file that depicts a binary object file in an ASCII representation. DINK32 can download either srec or binary files. See Appendix D in the *DINK32 User's Manual*.
- Sandpoint—Evaluation board for Freescale MPC6xx, 7xx, 74xx, 82xx processors. This board does not include an internal ethernet card.
- Target—Machine used with the kernel. For this application note, the target machine is either the Sandpoint or MVP machine.
- TFTP—Trivial File Transfer Protocol is a simple form of the FTP that features UDP and provides no security features.
- UDP—User DatagramProtocol, a communications protocol for the Internet network layer, transport layer, and session layer, which makes it possibe to send a datagram message from one computer to an application running in another computer.
- User and root type—Root user has super user permission to install RPMs, modify, create, and delete all files in all directories, mount and unmount file systems, start and end processes, and generally do anything on a Linux system. Unfortunately, this makes it easy to destroy the Linux development system and render it useless. For that reason, login as a root user only for those activities that require root permissions, and revert to normal user for all other activities, including building the Linux kernel. However, all the activities described in this paper require root permission, so care should be exercised in making these recommended changes.
- xinet, xinetd—xinet is the Internet service performed by xinetd, a daemon that monitors ethernet traffic to identify incoming traffic destined for the IP address on which it resides. This service and daemon is supported by Red Hat Linux V7.2 or Yellow Dog Linux V2.1.

- Yellow Dog Linux™—Product of Terra Soft Solutions, Inc., designer and distributor of of desktop Linux sources and solutions.

Part II General Methodology

The instructions in this application note are identical for Red Hat Linux V7.2, Mandrake Linux V7.0 (similar to Red Hat V7.0), and Yellow Dog Linux V2.1, except as noted. Much of this material is based on *Hard Hat Linux 2.0 Journeyman Edition Installation and Setup Guide* (supplied with the MontaVista distribution package).

Included in this section is an overview of the boot process and the steps required to build the NFS BOOTP/DHCP server.

2.1 Overview

This overview shows the interaction between a client and the NFS BOOTP/DHCP server, during the boot process. The process for BOOTP request is defined in the following steps. The BOOTP request routine for a known client is demonstrated in Figure 1.

- A boot record is loaded on the client (Sandpoint or MVP), initializes an ethernet driver, makes a connection, and broadcasts a BOOTP request which contains the client's MAC address.
- The BOOTP request is sent to a subnet and, if no BOOTP server is listening, the client times out.
- If a BOOTP server is listening, it responds. In this case, we are using a DHCP server.
- The DHCP server initially accepts the BOOTP request and compares the client's MAC address to a list of known valid clients (dhcpd.conf file).
- If it is not a valid client, the server refuses the request.
- If it is a valid client, the server responds with an IP address.

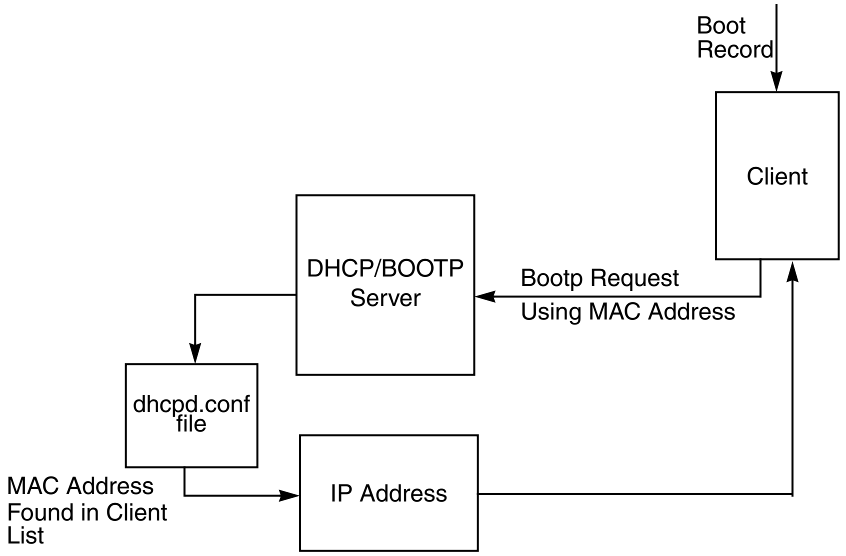

Figure 1. BOOTP Request Routine For Known Client

The following steps summarize the balance of the boot process. Figure 2 represents the boot record and root file system pointer request routine.

- The client then configures itself, using the IP address that was sent from the BOOTP server.
- The client then requests a pointer to a boot record and a root system file.
- The server responds with this information.
- Using the the pointer, the client may download the boot record, and continues the boot process.
- The client connects to the root file system.

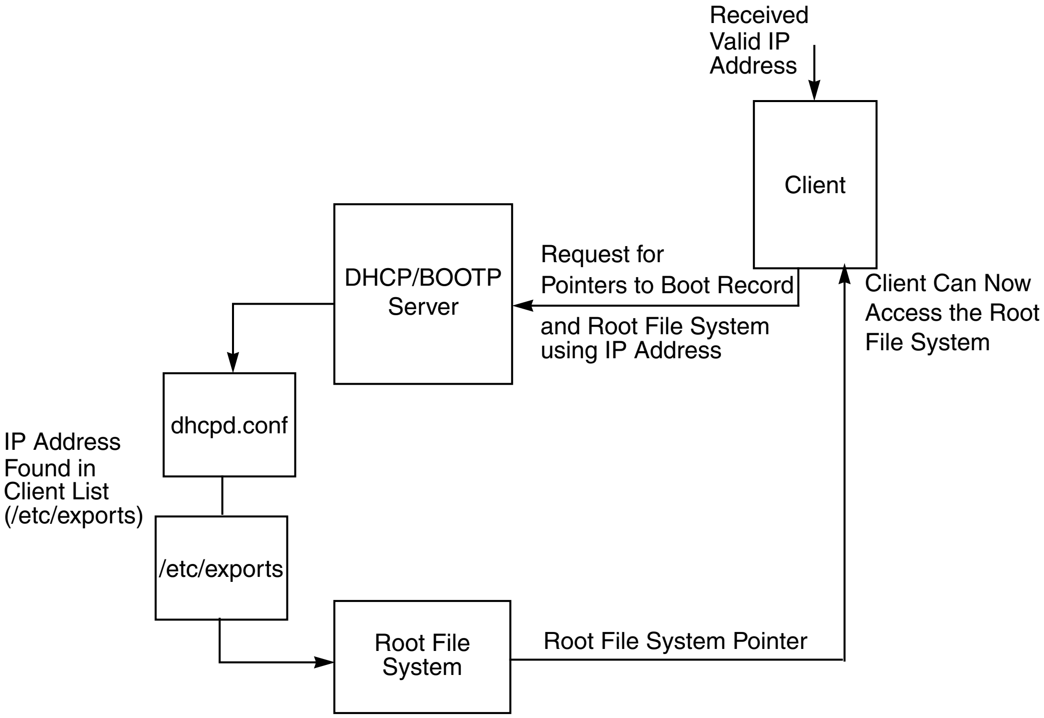

Figure 2. Boot Record and Root File System Pointer Request Routine

2.2 Required Server Steps

The following steps are required to build the NFS BOOTP/DHCP server. These steps are detailed in the sections that follow.

1. Install MontaVista CDK V2.0 with a target root system.
2. Build a kernel that uses an NFS boot on a host.
   See Application Note AN2222/D, *Porting Linux to the MPC8245.*
3. Read very carefully the cautionary note found in Part VI, “Dynamic Host Configuration Protocol (DHCP).”
4. Install, set up, and start a DHCP server.
5. Install, set up, and start a TFTP protocol.
6. Create a symbolic link to the boot kernel.
7. Set up NFS, includes exports file set up.
8. Set up a hostname.

Part III Server Machine

This section describes the server environment, identifying the software releases used, and details the Red Hat Package Manager packages (RPMs) that are required to build the server.

3.1 Server Environment

Two server platforms are available, the G4 Mac and the PC. The instructions shown here are identical regardless of which platform is used. The process defined in this document was executed using three different servers, RedHat Linux 7.2 on a PC (Pentium 166MHz) and Mandrake LinuxV7.0 on a PC (Pentium 166MHz); it was also executed on the G4 PPC using Yellow Dog Linux V2.1.

It is critical that, per the platform selected, the NFS connected boot image must include the object code for that specific target system. In this case, both target systems, that is, the Pentium 166 MHz PC, which does not support PowerPC architecture, and G4 PPC, which does implement PowerPC architecture. MontaVista CDK, V2.0 supplies this NFS connected boot image. Thus, the target system has PowerPC executables and can reside on either a non-PowerPC system or on a system using the PowerPC architecture.

To create a server machine, do one of the following:

- Install Linux on a PC, using either the RedHat or Mandrake distribution.
- Install LinuxPPC on a G4-running Mac OS, using the Yellow Dog distribution.

3.2 Required Red Hat Package Manager Packages (RPMs)

Certain RPMs must be installed depending on the version of Linux being used. The following sections indicate the RPMs and dependencies.

3.2.1 MontaVista CDK, V2.0

In order to build an srecord, use of zsrec instruction is required, that is only available from MontaVista and is included on the MontaVista release in the /common/ directory of the CDROM. See Part IV, “Installing the MontaVista CDK V2.0” for additional information. The instructions to build a kernel and the related srecord are found in AN2222/D, *Porting Linux to the MPC8245*, available on the Freescale web site, www.freescale.com.

- rpm-ihv hhl-zsrec-1.05.i386.rpm

3.2.2 Host System

Load the appropriate RPM, depending on the version of Linux being used. In this case, the version used was:

- dhcp-2.0pl5-8.i386.rpm

NOTE

This is the current version for Red Hat V7.2. You must have some version of DHCP installed.

- xinetd-2.3.3-1.i386.rpm

NOTE

> This is the current version for Red Hat V7.2. You must have some version of xinetd installed for Red Hat V7.2 and Yellow Dog V2.1.

- netkit_base-0.11-15mdk

NOTE

> This is the current version of inetd for Mandrake V7.0.

Part IV Installing the MontaVista CDK V2.0

Covered in this section is a summary of general information to assist in installing MontaVista Hard Hat Linux CDK V2.0, an overview of the corrections and differences to be considered, and directory locations.

4.1 General Information

You must install the PPC Cross Development Kit V2.0, which includes the target boot disk image.

Obtain the MontaVista Hard Hat Linux CDK V2.0 from the MontaVista web site, www.mvista.com. Install the PPC Linux kernel. For a host system that is based on a processor that implements the PowerPC Mac G4, you will use the native GNU tool chain. For host systems not based on the PowerPC architecture, install the PPC GNU cross tool chain.

In both cases, the user must produce a code kernel for the PowerPC architecture. A complete discussion on building the kernel is given in Application Note AN2222/D, *Porting Linux to the MPC8245*.

Follow the instructions supplied with the MontaVista distribution for the proper install. In this case, install the 82xx kernel and tools for a cross-hosted development, described in Chapter 2 of *Hard Hat Linux 2.0 Journeyman Edition Installation and Setup Guide* available from MontaVista Software.

4.2 Corrections and Differences

The MontaVista 2.0 script, /mnt/cdrom/bin/hhl-host-install (see *Hard Hat Linux 2.0 Journeyman Edition Installation and Setup Guide*,” page 7) is designed for the Red Hat V7.0 and will not work on other versions of Red Hat. This document is based on Red Hat V7.2.

An easy fix for any Linux development system is to change the release name temporarily. For Red Hat Linux V7.2, edit the file /etc/redhat-release, changing one line only. Change *Red Hat Linux release 7.2 (Enigma)* to *Red Hat Linux release 7.0 (Enigma).*

Be certain to change the line back after installing Hard Hat Linux.

Both the Mandrake release and the Yellow Dog release also have this /etc/redhat-release file. Hence, this install script works with any Red Hat, Mandrake, and Yellow Dog.

NOTE

> This document is written for the Red Hat Release 7.2. All differences between this, Mandrake (Red Hat V7.0) and Yellow Dog V2.1 are indicated in the appropriate sections.

The complete command to install the MontaVista target directory is:

```
/mnt/cdrom/bin/hhl-host-install --install --lspfreescale-sandpoint-82xx
```

This procedure has been tested on Red Hat V7.2, Mandrake V7.0, and Yellow Dog V2.1.

NOTE

> The Yellow Dog host will get some errors during this install. These errors can be ignored. However, the root file system will be mounted as read-only on Sandpoint.
>
> Red Hat V7.2 and V7.0 does not get errors and the root file system is mounted as read/write, and that is the expected result.

4.3 Directory Locations

Once installed, all the code and tools will reside starting in the directory, /opt.

- /opt/hardhat/host
  - — contains some important cross tools, specifically, the zsrec application that converts elf files into srecord files. This directory is only created on non PPC systems.
- /opt/hardhat/devkit/ppc/82xx/bin
  - — contains all the cross compiler tool chain for systems not based on the PowerPC architecture.
- /opt/hardhat/devkit/ppc/82xx/target
  - — contains a linux target system that can be used to build a Linux hard drive. This is the directory that is used for the NFS remote boot.
- /opt/hardhat/devkit/lsp
  - — contains all the kernel code for specific architectures. You want the freescale-sandpoint distribution. Specifically, you will find this distribution at /opt/hardhat/devkit/lsp/freescale-sandpoint/linux-2.4.2_hhl20.

/opt is owned by root and therefore you must have root permission to make any changes to these directories

Part V Linux Boot Methods

This section outlines common methods of booting Linux, depending on the hardware used.

5.1 Hard Drive Boot

If you are using a hard drive for your system, see Part IV in Application Note AN2222/D, *Porting Linux to the MPC8245*.

5.2 RAM Disk

If you are using a RAM disk for your system, see Part V in Applicaton Note AN2222/D, *Porting Linux to the MPC8245*.

5.3 NFS Remote Boot

The NFS remote boot from a server onthe network is the subject of this paper.

Part VI Dynamic Host Configuration Protocol (DHCP)

This section outlines how to set up and configure dynamic host configuration protocol (DHCP) and includes the necessary code for the DHCP daemon configuration file, a line-by-line explanation of that code, and instructions for restarting the DHCP daemon.

CAUTION

Do not copy IP addresses verbatim from this document; use a correct IP address for your network.

Do not have more than one DHCP server active per network.

Refer to Part VI, “Dynamic Host Configuration Protocol (DHCP).”

6.1 Set Up DHCP

Before configuring the DHCP, ensure that it is installed. Select a method based on the version of Linux being used.

- Red Hat V7.x and Yellow Dog V2.1: Ensure that the file, “/var/lib/dhcp/dhcpd.leases,” exits or create an empty file with the touch command.
- Red Hat V6.x: Ensure that the file, “/var/state/dhcp/dhcpd.leases,” exits or create an empty file with the touch command.

6.2 Configuring DHCP

The following sections address configuring the DHCP.

6.2.1 General DHCP Documentation

To configure a DHCP server, first create a configuration file. Composed of two types of statements, parameters and declarations, the configuration file stores network information for clients including:

- Parameters determines the following:
  - What network configuration options to send a client.
  - If a task should be performed
  - How the task is to be performed.
- Declarations describe the network and its clients. Declarations also provide addresses for the clients. There are two distinct declaration options:
  - Global declarations apply to all network clients.
  - An individual option declaration applies to a specific client system.

Consider the following points when developing a configuration file:

- Simplify formatting by using extra tabs or blank lines in the configuration file.
- Lines beginning with a hash mark (#) are comments.
- Key words are not case sensitive.
- Parameters beginning with the “option” keyword are required to configure DHCP options.
- Parameter statements not beginning with the “option” keyword either control the DHCP server behavior or provide non-optional values.
- Global parameters are those parameter declarations (including options) found prior to a section surrounded by curly brackets ({}) and apply to all sections that follow.

Refer to Chapter 12 of *The Official Red Hat Linux Customization Guide* found on http://www.redhat.com for more information.

NOTE

> Changes to a configuration file do not take effect until the DHCP daemon is restarted using the following command:
>
> ```
> service dhcpd restart
> ```

6.2.2 DHCP Daemon Configuration File

Before making any changes, please read the cautionary note found in the introduction to Part VI, “Dynamic Host Configuration Protocol (DHCP).”

Edit the /etc/dhcpd.conf file, using the following dhcpd.conf file and inserting the appropriate IP addresses:

```
allow bootp;
subnet 163.11.104.0 netmask 255.255.255.0 {
  option routers 163.11.104.254;
group {
    host realtek1 {
      hardware ethernet 00:40:c7:87:50:b2;
      fixed-address 163.11.104.163;
      filename "vmlinuz-freescale-sandpoint";
      option root-path "/opt/hardhat/devkit/ppc/82xx/target";
    }
    host mvp1 {
      hardware ethernet fe:ff:ff:00:00:01;
      fixed-address 163.11.104.164;
      filename "vmlinuz-freescale-sandpoint";
      option root-path "/opt/hardhat/devkit/ppc/82xx/target";
    }
  }
}
```

6.2.2.1 Line by Line Explanation

The DHCP daemon configuration file is explained here.

1. `allow bootp;`
   The NFS booting Linux client will issue a BOOTP request to the network. This server will respond to the request by sending a valid IP address, or refusing an unknown requester.
2. `subnet 163.11.104.0 netmask 255.255.255.0 {`
   This line denotes the subnet for the requesters, in this case, any requester on the triplet net 163.11.104. The netmask indicates to listen to all nets. Specify an appropriate subnet for your installation.
3. `option routers 163.11.104.254;`
   This is the address of the router for this subnet; specify a correct router for your network.
4. `group {`
   This line indicates the beginning of this group of clients.
5. `host realtek1 {`
   In this case there are two clients. This is the first client and its name is, realtek1, because we are using a Realtek ethernet card on a sandpoint since there is no internal ethernet interface. It could be any name you desire, however, it should correspond to the client's name. Even though this is setting up a client connection, this file calls it a host.
6. `hardware ethernet 00:40:c7:87:50:b2;`
   This is the MAC address of the client. When the client sends a BOOTP request on this subnet, it sends it's MAC address. Specify the correct MAC address for your client.
7. `fixed-address 163.11.104.163;`
   When the client with this specified MAC address issues the BOOTP request, this server will respond with this IP address. The client will then use this IP address. Specify an appropriate IP address here.
8. `filename "vmlinuz-freescale-sandpoint";`
   This is the initial boot file, that the client will use. This file is in /opt/hardhat/devkit/ppc/82xx/target/boot.

   NOTE

   Hard Hat Linux 2.0 Journeyman Edition Installation and Setup Guide, page 103 lists the available boot file names.
9. `option root-path "/opt/hardhat/devkit/ppc/82xx/target";`
   This line specifies the path for the clients NFS mount. All files normally in a mount root file system are here on the hard drive of the server.
10. `}`
    Close host realtek1 record. This ends the configuration for the first client.
11. `host mvp1 {`
    This is the second client, called mvp1, which uses an mvp board with the Galileo network interface.
12. `hardware ethernet fe:ff:ff:00:00:01;`
    This is the second client's MAC address.
13. `fixed-address 163.11.104.164;`
    This is the IP that will be assigned to this client.

14. `filename "vmlinuz-freescale-sandpoint;`
    Again, this is the initial boot file.
15. `option root-path "/opt/hardhat/devkit/ppc/82xx/target";`
    Again, this is the path for the NFS mount.

NOTE

Note that both clients are now using the same system file path. If either client changes anything in these directories, it will affect both clients. This is controlled in the file /etc/exports, see 8.1, “Set Up the Exports File”.

16. }
    Close host mvp1 record. This ends the configuration for the second client.
17. }
    Close group.
18. }
    Close subnet.

NOTE

`default-lease-time` and `max-lease-time` are not needed here, they are assumed to be infinite, that is, the client keeps the lease until it releases it.

This server will only serve two clients, MAC addresses 00:40:c7:87:50:b2 and fe:ff:ff:00:00:01. Any other client requester will be refused.

6.2.3 Restarting the DHCP Daemon

Assuming the DHCP daemon is running, it is automatically started up when linux boots, it is now necessary to restart the daemon. The daemon only reads its configuration file, dhcpd.conf, when starting.

Issue one of the following commands:

```
/etc/rc.d/init.d/dhcpd restart
```

or

```
service dhcpd restart
```

As a result, the following should appear on the console.

```
Shutting down dhcpd: [  OK  ]
Starting dhcpd: [  OK  ]
```

Part VII Trivial File Transfer Protocol (TFTP)

This section provides the information needed to set up the TFTP, including Linux release specific instructions, the TFTP file code, an explanation of the TFTP file code, the commands needed to create a symbolic link to the boot file, and TFTP restart instructions.

7.1 Mandrake V7.0 (Red Hat V7.0)

Edit /etc/inetd.conf and uncomment the TFTP line, as there is no TFTP file.

```
# Tftp service is provided primarily for booting.  Most sites
# run this only on machines acting as "boot servers." Do not uncomment
# this unless you *need* it.
#
tftp    dgram   udp wait    root    /usr/sbin/tcpd  in.tftpd
```

7.2 Red Hat V7.2 and Yellow Dog V2.1

There is no /etc/inetd.conf, instead the same functionality is created in the following manner.

- Create or edit this file, /etc/xinetd.d/tftp

7.3 TFTP File

Below is the code for the TFTP file being used.

```
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot
        disable                 = no
}
```

7.3.1 Line by Line Explanation

The TFTP file code is explained here.

1. `service tftp`
   Required first line to specify the service
2. `{`
   Indicate start of information.
3. `socket_type            = dgram`
   Using datagram service
4. `protocol               = udp`
   The protocol is udp
5. `wait                   = yes`
   wait for service
6. `user                   = root`
   This service must run in root
7. `server                 = /usr/sbin/in.tftpd`
   We chose to use our own local server, using the local TFTP service. The MontaVista target also contains a TFTP service, which is suggested in *Hard Hat Linux 2.0 Journeyman Edition Installation and Setup Guide*. The alternative line to use the service supplied by MontaVista is:
   `server                 = /opt/hardhat/host/bin/in.tftpd`

8. `server_args               = -s /tftpboot`
   Our local tftp requires the -s argument. The MontaVista target does not requres the -s argument so its line would be:
   `server_args               = /tftpboot /*`
9. `disable                   = no`
   Do not disable this service, that is, allow remote access.
10. }
   Indicates the end of information

7.4 Create a Symbolic Link

TFTP can only download files in the TFTP boot directory. Therefore, in order for the client to download an optional kernel, a link to this kernel must be placed in the TFTP boot directory.

Create a symbolic link in the TFTP boot directory to the boot file with the following commands:

```
cd /tftpboot
ln -s /opt/hardhat/devkit/ppc/82xx/target/boot/vmlinux-freescale-sandpoint
vmlinux-freescale-sandpoint
```

To see the link, execute this command:

```
ls -l vmlinux-freescale-sandpoint
```

As a result the link is displayed, prefixed by the letter “l,” as highlighted in this example:

```
lrwxrwxrwx        1  root          root                      67  Feb  21  10:48
vmlinux-freescale-sandpoint                                                      ->
/opt/hardhat/devkit/ppc/82xx/target/boot/vmlinux-freescale-sandpoint
```

7.5 Restart TFTP

Per the Linux release being used, issue the following commands as needed.

7.5.1 Mandrake V7.0 (Red Hat V7.0)

Use any of the following commands to restart TFTP.

```
killall -HUP inetd
```

or

```
/etc/rc.d/init.d/inet stop
/etc/rc.d/init.d/inet start
```

NOTE

Note the stop/start version is inet, not inetd.

7.5.2 Red Hat V7.2 and Yellow Dog V2.1

Use any of the following commands to restart TFTP.

```
killall -HUP xinetd
```

or

```
/etc/init.d/xinetd stop
/etc/init.d/xinetd start
```

or

```
service xinetd restart
```

Part VIII Set up Network File System

This section details the steps to setting up the NFS, including setting up the exports file and controlling the NFS daemon.

8.1 Set Up the Exports File

The server refers to the exports file to determine which requesters, via their IP address, are qualified to access, that is, export, this root file system. Edit the /etc/exports file by adding the appropriate line described in Section 8.1.1, "Valid Line for Red Hat V7.2" or Section 8.1.2, "Recommended Line."

8.1.1 Valid Line for Red Hat V7.2

The following line may be used for Red Hat Linux V7.2:

```
/opt/hardhat/devkit/ppc/82xx/target
163.11.104.*(rw,no_root_squash,no_all_squash)
```

8.1.2 Recommended Line

The following form works with all three Linux releases, for Red Hat V7.2, MontaVista V7.0, and Yellow Dog V2.1. This is the recommended version:

```
/opt/hardhat/devkit/ppc/82xx/target
163.11.104.0/24(rw,no_root_squash,no_all_squash)
```

or

```
/opt/hardhat/devkit/ppc/82xx/target
163.11.104.0/255.255.255.0(rw,no_root_squash,no_all_squash)
```

This allows all clients on the subnet 163.11.104 to have root access to the file system. Which means that any changes made by those clients will physically change the client root file system that resides on the server. This does not mean that the server's root file system is changed, only the client's file system is affected. However, all clients that have access to this root file system can change it and there is no synchronization between them.

8.2 Controlling the NFS Daemon

The following sections describe how to restart the NFS daemon, run the daemon with each boot, and synchronize the daemon.

8.2.1 Restart the Daemon

This section provides instructions on how to check the portmapper status, start the portmapper, and start, restart, and stop the NFS daemon.

8.2.1.1 Check the Portmapper Status

When restarting the NFS daemon, it is necessary to have the portmap running. If it is not running, the port mapper must be started, before starting or restarting NFS daemon. Use the following command to determine the status of the portmapper.

```
service portmap status
```

If portmap is not running then start it using the following command.

```
service portmap start
```

8.2.1.2 Start or Restart NFS Daemon

Use the following instructions to restart, stop or start NFS daemon.

```
/etc/rc.d/init.d/NFS restart
```

or

```
/etc/rc.d/init.d/NFS stop
/etc/rc.d/init.d/NFS start
```

or

```
service NFS restart
```

8.2.2 Run Daemon with Each Boot

To run the NFS daemon on every boot, use the following:

```
/sbin/chkconfig NFS on
```

8.2.3 Synchronize the NFS Daemon

The following command synchronizes the NFS daemon.

```
/usr/sbin/exportfs -ra
```

Part IX Set Up resolv.conf File

Before making any changes please read the cautionary note found in Part VI, “Dynamic Host Configuration Protocol (DHCP).”

Once everything is working and the client is booted via NFS, then all network facilities will be available. However, domain name resolution will not work until the target has the resolv.conf file correctly built.

NOTE

> The resolv.conf file must be built in the target (client) root file system, which is /opt/hardhat/devkit/ppc/82xx/target/etc. In this location, this file affects the client, not the server.
>
> There is also a resolv.conf file in the host (server) root file system, which only affects the server, not the client.

In the following example of the resolv.conf file, be certain to supply the correct IP address, domain, and search path names.

```
nameserver 192.55.22.4
nameserver 192.5.249.4
nameserver 192.5.248.76
domain sps.mot.com
search sps.mot.com
```

9.1 Line by Line Explanation

The resolv.conf file code is explained here.

1. `nameserver 192.55.22.4`
   IP addresses of a name server
2. `nameserver 192.5.249.4`
   first alternate IP address of a name server
3. `nameserver 192.5.248.76`
   second alternate IP address of a name server
4. `domain sps.mot.com`
   domain name for this network
5. `search sps.mot.com`
   search name for machines on this network.

Part X Other Problems

If the sandpoint or MVP boards can not start a shell, then it is necessary to add this link to the /opt/hardhat/devkit/ppc/82xx/target/bin directory.

```
ln -s bash sh
```

Part XI Host Ethernet Diagnostic Commands

The following host ethernet diagnostic commands permit the user to check the gateway and network set up, view network traffic, and access the message log.

11.0.1 Gateway and Network Setup Status Check

Use the following command to see the status of your gateway and network setup as determined from the hostname file.

```
netstat -r
```

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
163.11.104.0    *               255.255.255.0   U        40 0          0 eth0
127.0.0.0       *               255.0.0.0       U        40 0          0 lo
default         163.11.104.254  0.0.0.0         UG       40 0          0 eth0
```

11.0.2 Network Traffic Viewer

The following command sets up a promiscuous bit in the network mask so that all traffic can be seen.

```
tcpdump -n -i eth0 arp
tcpdump: listening on eth0
10:59:59.380606 arp who-has 163.11.104.44 tell 163.11.104.253
10:59:59.380606 arp who-has 163.11.104.44 tell 163.11.104.253
11:00:01.400606 arp who-has 163.11.104.44 tell 163.11.104.253
11:00:01.400606 arp who-has 163.11.104.44 tell 163.11.104.253
```

11.0.3 Message Log

All messages produced by network activity and starting and stopping services are stored in the /var/log/messages file.This is especially useful if you need to determine the MAC address of one of your clients. You can inspect the traffic and look for BOOTP requests. In the example shown below, there is a BOOTP request from MAC address 00:40:c7:87:50:b2, which is the sandpoint “realtek” client. If you are setting up BOOTP clients in the dhcpd.conf file and you do not know the MAC address of your client, then let the client issue BOOTP requests, which go unanswered, and look at the messages file to see the MAC address.

```
tail -10 /var/log/messages
Feb 26 10:56:39 appslab2 su(pam_unix)[26101]: session opened for user root by
maurie(uid=500)
Feb 26 10:59:59 appslab2 kernel: eth0: Setting promiscuous mode.
Feb 26 10:59:59 appslab2 kernel: device eth0 entered promiscuous mode
Feb 26 11:00:04 appslab2 kernel: device eth0 left promiscuous mode
Feb 26 11:41:16 appslab2 dhcpd: BOOTREQUEST from 00:40:c7:87:50:b2 via eth0
Feb 26 11:41:16 appslab2 dhcpd: BOOTREPLY for 163.11.104.163 to realtek1
(00:40:
c7:87:50:b2) via eth0
Feb 26 11:41:17 appslab2 rpc.mountd: authenticated mount request from
163.11.104
.163:800 for /opt/hardhat/devkit/ppc/82xx/target
(/opt/hardhat/devkit/ppc/82xx/target)
```

Part XII Starting Linux with the Executable

Part XII, “Starting Linux with the Executable” indicates where instructions can be found to download the executable, gives examples of the splash screens, and provides a sample Sandpoint session.

12.1 Downloading the Executable

See AN2222/D, *Porting Linux to the MPC8245* for information on downloading the executable and starting the NFS version of Linux on the Sandpoint or MVP board.

12.2 Splash Screen for MVP

The splash screen is shown here indicates that NFS was enabled during configuration, the Galileo ethernet device was specified, and BOOTP was enabled.

```
Baud rate changing to 115200...

enable port 74/75
MIRQ
GetIdentity -- skipped for now.
Memory Enabled: [ 128MB at CL=6 ]
Caches Enabled: [ L1-ICache L1-DCache ]
Register Inits: [ 32 GPRs, 32 FPRs, 155 SPRs, 32 VECs ]
Assembler Init: [ 895 opcodes ]

         ######  ### #     # #   #  #####   #####
         #     #  #  ##    # #  #  #     # #     #
         #     #  #  # #   # # #         #       #
         #     #  #  #  #  # ##     #####   #####
         #     #  #  #   # # # #         # #
         #     #  #  #    ## #  #  #     # #
         ######  ### #     # #   #  #####  #######

        (        (    (  ( (AltiVec) )  )    )        )

Version : XX.X, Metaware Build
  Released : Interim Release:  Built on Apr  4 2002 23:24:45
Written by : Motorola's RISC Applications Group, Austin, TX
    System : MVP Multi-processor V'ger Platform, 60XBus
 Processor : MPC7450 V2.1 @ 600 MHz, Memory @ 100 MHz
    Memory : 128MB at 6/1/1/1

Copyright Inc. 1993-2002
Refer to `history.c' for release info, changes, errata and fixes.

MultiProcessor Status:
  CPU0: MPC7450-600 MHz active
  CPU1: MPC7450-600 MHz active
DINK32[MPC7450 #0] {1} >>dl -k -b -o 900000
DINK32[MPC7450 #0] {2} >>go 900000
```

```
MVP
lmrcCd
memspeed: 00000064
delay test
cpu_flag: DEADF00F
parking cpu1 in bootloader
LM
loaded at:     00900000 009B05F0
relocated to:  00800000 008B05F0
avail ram:     00400000 00800000

Linux/PPC load:
Uncompressing Linux...done.
signalling cpu1 to park in kernel code
cpu1 Parking
cpu_flag: 00002000
Now booting the kernel
 :
:hhrhp
setup_arch: enter
setup_arch: bootmem
mvp_setup_arch: enter
mvp_setup_arch: L2CR

mvp_setup_arch: find_bridges
gt64260_base already set, skipping ioremap
mvp_setup_arch: exit
arch: exit
gt64260_init_irq: enter
gt64260_init_irq: GPP -> levelint
gt64260_init_irq: exit
gt64260mpsc_console_init: enter
gt_console_setup: enter
gt_console_setup: exit
Memory BAT mapping: BAT2=32Mb, BAT3=0Mb, residual: 0Mb
Total memory = 32MB; using 64kB for hash table (at c01e0000)
Linux version 2.4.19-pre4 (maurie@appslab2.sps.mot.com) (gcc version 2.95.3
20010315 (release/MontaVista)) #3 Thu Apr 4 09:40:08 CST 2002
cpu 0 l2cr now: 0x80000000
hid0: 0x8411c0bc
l2cr: 0x80000000
l3cr: 0x       0
CNTL0: 0x00007777 CNTL1: 0x00000000 CNTL2: 0x00888888 CNTL3: 0x00090000
GPP_IO   : 0x01800000 GPP_LEVEL: 0x000002c6 GPP_VALUE: 0xed80fd36
GPP_INTR_CAUSE: 0xfd7fffff GPP_INTR_MASK: 0x00000000
CNTL0: 0x00007777 CNTL1: 0x00000000 CNTL2: 0x00888888 CNTL3: 0x00000000
GPP_IO   : 0x81800020 GPP_LEVEL: 0x0100ffc0 GPP_VALUE: 0xed800030
GPP_INTR_CAUSE: 0xfd7fffff GPP_INTR_MASK: 0x0000ffc0
MVP (Freescale Vger Platform) Evaluation Board
MVP port (C) 2001 MontaVista Software, Inc. (source@mvista.com)
On node 0 totalpages: 8192
zone(0): 8192 pages.
zone(1): 0 pages.
zone(2): 0 pages.
```

```
Kernel command line: console=ttyS0,115200 ip=on
time_init: WARNING: HARDCODED FREQUENCY
time_init: decrementer frequency = 25.000000 MHz
gt64260mpsc_console_init: exit
Calibrating delay loop... 599.65 BogoMIPS
Memory: 30264k available (1160k kernel code, 652k data, 72k init, 0k highmem)
Dentry-cache hash table entries: 4096 (order: 3, 32768 bytes)
Inode-cache hash table entries: 2048 (order: 2, 16384 bytes)
Mount-cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer-cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 8192 (order: 3, 32768 bytes)
POSIX conformance testing by UNIFIX
PCI: Probing PCI hardware
Activating ISA DMA hang workarounds.

Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
Starting kswapd
JFFS2 version 2.1. (C) 2001 Red Hat, Inc., designed by Axis Communications AB.
pty: 256 Unix98 ptys configured
gt_mpsc_init: exit
block: 64 slots per queue, batch=16
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
userflash: 0x1000000 at 0xfe000000
 Amd/Fujitsu Extended Query Table v1.1 at 0x0040
number of CFI chips: 1
Creating 3 MTD partitions on "MVP User flash":
0x00000000-0x00e00000 : "fs"
0x00e00000-0x00f00000 : "kernel"
0x00f00000-0x01000000 : "Dink32"
 bootflash: 0x1000000 at 0xff000000
CFI: Found no MVP Boot flash device at location zero
map probe failed for bootflash
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 2048)
gt64260_eth_open : Assigned IRQ 32 to gt64260_eth0
eth0: link state:
  GT:100:nLink:HD:nFC
 mii: 10:nLink:HD:nFC ANnc:AN
gt64260_eth_open : Assigned IRQ 33 to gt64260_eth1
eth1: MII said 0, GT said 9, restarting autoneg
eth1: link state:
  GT:100: Link:HD:nFC
eth1: changed link status to DOWN
 mii: 10:nLink:HD:nFC ANnc:AN
Sending DHCP requests .eth1: changed link status to UP
NETDEV WATCHDOG: eth0: transmit timed out
., OK
IP-Config: Got DHCP answer from 163.11.104.162, my address is 163.11.104.164
```

```
IP-Config: Complete:
      device=eth1, addr=163.11.104.164, mask=255.255.255.0, gw=163.11.104.254,
     host=163.11.104.164, domain=, nis-domain=(none),
                      bootserver=163.11.104.162,    rootserver=163.11.104.162,
rootpath=/opt/hardhat/devkit/ppc/82xx/target
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
Looking up port of RPC 100003/2 on 163.11.104.162
Looking up port of RPC 100005/1 on 163.11.104.162
VFS: Mounted root (nfs filesystem) readonly.
Freeing unused kernel memory: 72k init
INIT: version 2.78 booting
Activating swap...
Checking all file systems...
Parallelizing fsck version 1.19 (13-Jul-2000)
Mounting local filesystems...
not mounted anything
Cleaning: /etc/network/ifstate.
Setting up IP spoofing protection: rp_filter.
Disable TCP/IP Explicit Congestion Notification: done.
Configuring network interfaces: done.
Starting portmap daemon: portmap.
Cleaning: /tmp /var/lock /var/run.
INIT: Entering runlevel: 2
Starting system log daemon: syslogd klogd.
Starting internet superserver: inetd.

Freescale Applications team in conjuction with
MontaVista Software's Hard Hat Linux 2.0

163.11.104.164 login: maurie
Password:
Last login: Thu Jan  1 00:12:09 1970 from 163.11.105.183 on pts/0
Linux 163.11.104.164 2.4.19-pre4 #3 Thu Apr 4 09:40:08 CST 2002 ppc unknown

Welcome to MontaVista Software's Hard Hat Linux.
in conjuction with Freescale applications team

maurie@163.11.104.164:~$ ls
dhrystone_cross_ppc  dhrystone_cross_ppc.tar  env  mt  vdink32  vdink32_old
maurie@163.11.104.164:~$ users
maurie mvp1
maurie@163.11.104.164:~$ su -
Password:
root@163.11.104.164:~# shutdown -h now

Broadcast message from root (console) Thu Jan  1 00:20:50 1970...

The system is going down for system halt NOW !!
INIT: Switching to runlevel: 0
INIT: Sending processes the TERM signal
INIT: Sending processes the KILL signalStopping portmap daemon: portmap.
Stopping internet superserver: inetd.
Stopping system log daemon: klogd syslogd.
Sending all processes the TERM signal... done.
Sending all processes the KILL signal... done.
Unmounting remote filesystems... done.
Deactivating swap... done.
Unmounting local filesystems... done.
Power down.
```

12.2.1 MVP Sample Session

The following text is from an MVP sample session. Input commands are shown in bold text.

```
Connected to appslab4.sps.mot.com.
Escape character is '^]'.

MontaVista Software's Hard Hat Linux 2.0
Linux/ppc 2.4.19-pre4

163.11.104.164 login: mvp1
Password:
Last login: Thu Jan  1 00:18:26 1970 from appslab1.sps.mot.com on pts/0
Linux 163.11.104.164 2.4.19-pre4 #3 Thu Apr 4 09:40:08 CST 2002 ppc unknown

Welcome to MontaVista Software's Hard Hat Linux.
in conjuction with Freescale applications team

mvp1@163.11.104.164:~$ whoami
mvp1
mvp1@163.11.104.164:~$ ls
dhrystone_cross_ppc  dhrystone_cross_ppc.tar  env  mt  vdink32  vdink32_old
mvp1@163.11.104.164:~$
Broadcast message from root (console) Thu Jan  1 00:20:50 1970...

The system is going down for system halt NOW !!
Connection closed by foreign host.
[maurie@appsla
```

12.3 Splash Screen for Sandpoint

The splash screen is shown here indicates that NFS was enabled during configuration, the Realtek 8139 ethernet card was specified, and BOOTP was enabled.

```
Duart Initialized...
Skipping environment variables setup...
Memory Enabled: [ 64MB at CL=3 ]
Caches Enabled: [ L1-ICache L1-DCache ]
Register Inits: [ 32 GPRs, 32 FPRs, 155 SPRs ]
Assembler Init: [ 895 opcodes ]

         ######  ### #     # #   #  ######  ######
         #     #  #  ##    # #  #  #     # #     #
         #     #  #  # #   # # #         #       #
         #     #  #  #  #  # ##     ######  ######
         #     #  #  #   # # # #         # #
         #     #  #  #    ## #  #  #     # #
         ######  ### #     # #   #  ###### #######


Version : 12.3, Metaware Build
  Released : October 25, 2001:
Written by : Motorola’s RISC Applications Group, Austin, TX
    System : Sandpoint X3 with Unity (MPMC8240)
 Processor : MPC8240 V1.1 @ 250 MHz, Memory @ 100 MHz
    Memory : Map B (CHRP) 64MB at 3/1/1/1

Copyright 1993-2001
Refer to `history.c’ for release info, changes, errata and fixes.
DINK32 [MPC8240] >>sb -k 57600
 Baud rate changing to 57600...
DINK32 [MPC8240] >>dl -k
  Download from Keyboard Port
19803 lines received.
  Download complete.
DINK32 [MPC8240] >>sb -k 9600
Baud rate changing to 9600...
DINK32 [MPC8240] >>go 900000
loaded at:     00900000 009091B8
relocated to:  00800000 008091B8
zimage at:     00906000 0099968C
avail ram:     00400000 00800000

Linux/PPC load:
Uncompressing Linux...done.
Now booting the kernel
Total memory = 32MB; using 0kB for hash table (at 00000000)
Linux version 2.4.2_hhl20 (maurie@appslab2.sps.mot.com) (gcc version 2.95.3
2001
0315 (release/MontaVista)) #2 Wed Feb 20 16:56:48 CST 2002
Freescale SPS Sandpoint Test Platform
Sandpoint port (C) 2000, 2001 MontaVista Software, Inc. (source@mvista.com)
```

```
On node 0 totalpages: 8192
zone(0): 8192 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line:
OpenPIC Version 1.2 (1 CPUs and 24 IRQ sources) at f7fd0000
OpenPIC timer frequency is 100.000000 MHz
time_init: decrementer frequency = 24.753087 MHz
Calibrating delay loop... 164.65 BogoMIPS
Memory: 30580k available (1052k kernel code, 452k data, 88k init, 0k highmem)
Dentry-cache hash table entries: 4096 (order: 3, 32768 bytes)
Buffer-cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 8192 (order: 3, 32768 bytes)
Inode-cache hash table entries: 2048 (order: 2, 16384 bytes)
POSIX conformance testing by UNIFIX
PCI: Probing PCI hardware
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Starting kswapd v1.8
pty: 256 Unix98 ptys configured
block: queued sectors max/low 20245kB/6748kB, 64 slots per queue
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
Uniform Multi-Platform E-IDE driver Revision: 6.31
ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx
W82C105: IDE controller on PCI bus 00 dev 59
W82C105: chipset revision 5
W82C105: 100% native mode on irq 17
    ide0: BM-DMA at 0xbfffd0-0xbfffd7, BIOS settings: hda:pio, hdb:pio
SL82C105 command word: 5
IDE timing: 00000909, resetting to PIO0 timing
    ide1: BM-DMA at 0xbfffd8-0xbfffdf, BIOS settings: hdc:pio, hdd:pio
SL82C105 command word: 5
IDE timing: 000003e4, resetting to PIO0 timing
loop: loaded (max 8 devices)
Serial driver version 5.02 (2000-08-09) with MANY_PORTS SHARE_IRQ SERIAL_PCI
ena
bled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
ttyS01 at 0x02f8 (irq = 3) is a 16550A
8139too Fast Ethernet driver 0.9.13 loaded
eth0: RealTek RTL8139 Fast Ethernet at 0xbfffff00, 00:40:c7:87:50:b2, IRQ 19
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 2048)
Sending BOOTP requests.... OK
IP-Config: Got BOOTP answer from 163.11.104.162, my address is 163.11.104.163
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
Looking up port of RPC 100003/2 on 163.11.104.162
Looking up port of RPC 100005/2 on 163.11.104.162
VFS: Mounted root (NFS filesystem) readonly.
Freeing unused kernel memory: 88k init 4k openfirmware
modprobe: modprobe: Can’t open dependencies file
/lib/modules/2.4.2_hhl20/module
```

```
s.dep (No such file or directory)
INIT: version 2.78 booting
Activating swap...
Checking all file systems...
Parallelizing fsck version 1.19 (13-Jul-2000)
Mounting local filesystems...
not mounted anything
Cleaning: /etc/network/ifstate.
Setting up IP spoofing protection: rp_filter.
Configuring network interfaces: done.
Starting portmap daemon: portmap.
Cleaning: /tmp /var/lock /var/run.
INIT: Entering runlevel: 2
Starting system log daemon: syslogd klogd.
Starting internet superserver: inetd.

MontaVista Software’s Hard Hat Linux 2.0
163.11.104.163 login:
```

12.3.1 Sandpoint Sample Session

The following text is from a Sandpoint sample session. Input commands are shown in bold text.

```
163.11.104.163 login: maurie
Password:
Last login: Tue Nov 30 12:13:40 1999 from 163.11.105.124 on pts/0
Linux 163.11.104.163 2.4.2_hhl20 #2 Wed Feb 20 16:56:48 CST 2002 ppc unknown

Welcome to MontaVista Software’s Hard Hat Linux.

maurie@163.11.104.163:~$ cat /etc/hosts
127.0.0.1       localhost.localdomain   localhost
maurie@163.11.104.163:~$ cat /etc/exports
cat: /etc/exports: No such file or directory
maurie@163.11.104.163:~$ cat /etc/resolv.conf
#
# resolv.conf  This file is the resolver configuration file
# See resolver(5).
#
nameserver 192.55.22.4
nameserver 192.5.249.4
nameserver 192.5.248.76
domain sps.mot.com
search sps.mot.com
maurie@163.11.104.163:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
163.11.104.0    *               255.255.255.0   U        40 0          0 eth0
default         163.11.104.254  0.0.0.0         UG       40 0          0 eth0
maurie@163.11.104.163:~$ netstat -i
Kernel Interface table
Iface   MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500   0     4828      0      0      0     2433      0      0      0 BRU
lo     3904   0        0      0      0      0        0      0      0      0 LRU
```

```
maurie@163.11.104.163:~$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:sunrpc                *:*                     LISTEN
tcp        0      0 *:telnet                *:*                     LISTEN
udp        0      0 *:800                   *:*
udp        0      0 *:sunrpc                *:*
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node Path
root@163.11.104.163:~# cd /etc
root@163.11.104.163:/etc# cat /var/log/messages
Nov 30 05:20:52 163 syslogd 1.3-3: restart.
Nov 30 05:20:52 163 kernel: klogd 1.3-3, log source = /proc/kmsg started.
Nov 30 05:20:52 163 kernel: Cannot find map file.
Nov 30 05:20:52 163 kernel: No module symbols loaded.
Nov 30 05:20:52 163 kernel: Total memory = 32MB; using 0kB for hash table (at 
00
000000)
Nov 30 05:20:52 163 kernel: Linux version 2.4.2_hhl20 
(maurie@appslab2.sps.mot.com) (gcc version 2.95.3 20010315 
(release/MontaVista)) #2 Wed Feb 20 16:56:48 CST 2002
Nov 30 05:20:52 163 kernel: Freescale SPS Sandpoint Test Platform
Nov 30 05:20:52 163 kernel: Sandpoint port (C) 2000, 2001 MontaVista Software, 
Inc. (source@mvista.com)
```

Part XIII Revision History

Table 1 lists this document's significant changes and revisions.

Table 1. Document History

Rev. No.	Substantial Changes
1	Initial release
1.1	Nontechnical reformatting

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.
Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

freescale™
semiconductor